Active Learning Facilitates Success of Formal Methods in Practice

Victor V. Kuliamin, Vitaliy A. Omelchenko, and Olga L. Petrenko

Institute for System Programming of Russian Academy of Sciences (ISPRAS),
B. Communisticheskaya, 25, Moscow, Russia
{kuliamin,vitaliy,olga}@ispras.ru

Abstract. Software development does not consist only of technical activities, as it usually regarded. It also involves social and cognitive activities that are particularly important in complex projects. To make a software development project successful the staff involved should possess the necessary social and cognitive skills. The significance of such skills increases if some novel, advanced techniques, including formal methods, are used in the project. So, if we wish to leverage successful applications of formal methods, we need to train engineers and students not only in the techniques, but also in the social skills required to apply those techniques in real practice. Active learning methods seem to be very suitable for formal methods education, since they help to provide students with both kinds of necessary skills – the technical ones due to the content and the social and cognitive ones due to the form of the learning process.
Keywords: formal methods, teaching formal methods, active learning, cooperative learning, critical thinking pedagogical framework.

1 Introduction

Usually software development is depicted as a set of technical activities that are somehow related with economic issues such as resulting product sales and matching the project budget. But those issues are considered in computer science and software engineering courses as the minor ones and most attention is put to the techniques and method that can provide the best quality-effort ratio in software development projects. The tasks of determining the sufficient level of quality and the necessary level of effort are considered the topics of project management and seem to have little to do with the techniques used and work of single developer.

Practice shows that the things are slightly different in real projects. The decisions made by different members of development team on various steps and their abilities to communicate efficiently with all the other stakeholders and inside the team influence the project success or failure much more than the specific techniques used. The skills in decision making and effective communication are needed not only for the key staff as project managers, business analysts, and architects, but also for average developers. Many times during the project they face with necessity to make micro-decisions whether to implement something as

one component of two, or whether to solve certain task with the help of this algorithm or that one. And many times they communicate with management, architect or other developers and testers on details of the decisions made earlier.

The skills they need in this environment can be (not very strictly) partitioned into *cognitive skills* related with understanding of the new domains and requirements, constructing possible solutions without clear specifications, prioritizing the tasks, and choosing the most appropriate solutions, and *social skills* related with effective communication, accepting team goals, working in a team, coordinating your activities with needs of other people. Social skills also form the communication style of a person. They determine how he/she listens others and reacts on the things heard, how he/she discusses problems with other people and argue in favour of his/her viewpoint.

Those issues attract attention worldwide. In particular, they are addressed by the Memorandum on Lifelong Learning [1] produced by European Commission. Lifelong learning is defined there as all purposeful learning activity, undertaken on an ongoing basis with the aim of improving knowledge, skills and competence. This document underlines importance of such social skills as self-confidence, self-direction, and risk-taking and such cognitive ones as ability to learn, to adapt to changes, to acquire new skills rapidly, and to make sense of vast information flows for effective lifelong learning in modern digital economy era.

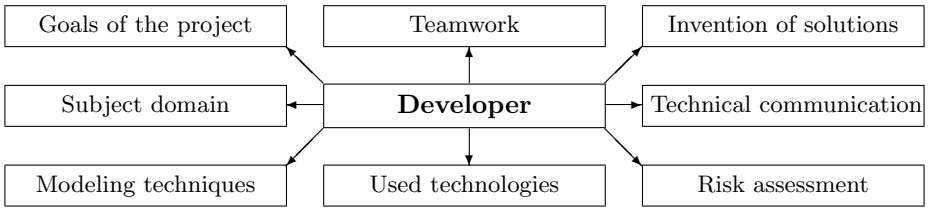

Fig. 1. Areas where modern developer should demonstrate worthy skills.

Usually those skills are considered to be important only for students, whose professional activity is related to communication with people – managers, teachers, social workers, etc. But when some advanced techniques, for example, formal methods, are used in a software development project, those skills become even more important for both the success of the project itself and the success of the case study. This is because the cognitive tasks in such a project are more complicated – we need not only to comprehend the domain and tasks to be solved, but also to express them in an unusual way needed to apply the advanced technique used effectively. In case of applications of formal methods the comprehension should be much deeper than for traditional development methods. And the number of micro-decisions to make is larger.

Serious social skills are needed to make this work accessible and actually useful for various stakeholders and other developers, which are not using formal

methods directly. The professionals in advanced development techniques should be conscious of people's unwillingness to deal with new things; they should be able to communicate with living people, for which formal proof is usually not the ultimate argument, while personal sympathies, sentiments, ambitions, and habits influence the decision very much. They will work in different organizations having different kinds of organizational culture and should be capable to avoid mistakes in presenting their results, pulling out the necessary information, and use the peculiarities of particular culture for greater project success.

So, to make possible successful use of formal methods in practice we need not only to teach the methodological base of them, but also to cultivate the necessary cognitive and social skills. The developed communication and cognitive skills should become as natural for student as the skills in theorem proving.

The point of this article is that modern active and collaborative learning methods [2, 3] are much more suitable instruments for studying formal methods than the traditional ones. The crucial reason for that is more effective cultivation of autonomous behavior, self-direction, decision making, communication skills, and ability to learn and to self-educate according to the changing needs in active and collaborative learning. Whereas classic education process mostly turns students to passive perception of information provided and produces well-informed people, who can hardly use their vast knowledge in rapidly changing context without permanent supervising.

Active learning is based on an idea that the teacher is no more a source of knowledge, he involves students into learning process, coordinates their own activities, force them to think independently on the course contents and try the approaches presented on practice. Classic exposition education style on the contrary is intended to make student to remember some piece of knowledge and to be able to reproduce it. Such style makes focus on attention and remembering the information, while all the aspects of the student's perception and emotional sphere should be involved in learning process to train social skills and self-direction.

Active learning methods stimulate students to active thinking and practicing during the learning process. Active learning methods are intended to help students in independent acquiring of skills and knowledge in the course of cogitative and practical activities. Some authors ([4]) distinguish active learning methods by the following characteristics.

- Forced activation of thinking, which make a student active independently of his/her wishes.
- Rather long period of involvement in learning activity justified by the need to make students steadily active for long time and escape their transitory activity.
- Self-dependent creative search for solutions and high level of motivation.

We do not try to describe all possible methods that can help in developing the necessary social and cognitive skills – it is impossible. We only present some techniques that push students to active thinking and practical activity during the learning process.

2 Active Learning in Formal Methods

Idea: **Test sequence generation with the help of an exploration of graph model of the system under test as an unknown environment**				
Propagandist	*Theorist*	*Technical expert*	*Manager*	*Investor*
Graph models of complex systems may be represented in a compact form	Explicit graph models provide more control on situation	More sound methods of quality control are urgent	Increase in software quality is promising	Can be used "as is" only in services on quality assurance of mission critical software
Much more complex situations can be tested automatically	Adaptive test sequences can be easily generated	Implicit graph models are harder to describe, at least for those who never met them before	No good tracing is provided between requirements and tests – why we need to test all those situations?	Lightweight method of graph model description should be invented
Quality of such testing is very high		Usability of the approach seem to need improvements	Special training is needed for testers to use this method	Useful plugins for widely used testing environments may be developed on the base of this approach
		Special GUI-based tool should be developed to guide the process of construction of such a model		

Fig. 2. Example of role-playing game results.

Each professional activity has its own crucial skills. Those skills can be technical, cognitive, or social. We determine the following social and cognitive skills as being very important in formal methods application in practice.

- Ability to hear what the opponents are saying.
- Ability to communicate effectively on professional topics. 'Effectively' here means that a person having such ability can get or give the necessary infor-

mation in rather short time without creating stress or tension for communication partners.
– Ability to adapt the one's knowledge to the changing situation.

This section enumerates several techniques that can be used in education process to facilitate development of those skills in conjunction with technical knowledge on formal methods. The techniques we consider below are role-playing game, clustering, brainstorming, and debates.

2.1 Role-playing Game

One of useful techniques is *role-playing game* (here we mean a learning technique, not its entertainment analogue!) or *simulation* [5, 6] Such games are used widely in the social science education, but much rarely in technical one. Role-playing game is an artificially constructed situation that models some important aspects of the real-life activity. During the game its participants paly certain roles and represent the corresponding views on the situation. The goal of such a game is to suggest possible solutions for the problems posed by the situation.

For example, if we wish to evaluate significance of some idea for student's work and its possible consequences, the following game can be organized.

The participants get different roles to make possible multi-purpose evaluation of the idea.

– 'Propagandist' advocates the idea, presents it, its perspectives and consequences in positive way.
– 'Theorist' considers relation of the idea to the fundamentals of the domain or other theoretical frameworks.
– 'Technical expert' evaluates practical significance of the idea, possibility to apply it in practice, and usability.
– 'Manager' tries to trace the consequences of the idea usage in practice, its impact on the industrial processes, workflows, and team work.
– 'Investor' estimates potential benefits of the idea applications and possibility to construct ready-to-use products on its base or integrate it in existing solutions.

Fig. 2 shows a result of role-playing game – evaluations of an idea from different viewpoints.

Role-playing games can increase students' interest to the domain studied, discover serious gaps in their knowledge, and train social skills. But they have significant restrictions, since a game is governed by strict rules and can not mirror real-life situations with their variability and dynamism. So, it is very hard to construct a set of games capable to cover major part of situations met in practice.

Role-playing game effectiveness can be increased by using *evolving tension* method. This method supposes psychological realignment of a person in response to artificially constructed or spontaneous emotionally-negative situation. As a result a person is trained to overcome stress situations successfully, do not afraid

them and even learn something positive from them, use them for self-training and self-education. Such a technique can be useful to train students in self-confidence and decision making skills.

2.2 Clustering

Active learning approach assumes not only new learning techniques and new criteria to evaluate students' work, but also new role of the teacher as a students' facilitator in their independent acquiring of new knowledge. Active learning techniques are targeted to force student's activity in dealing with piece of knowledge, analysing it, confronting and comparing it with other pieces. Teacher should not support 'say me what to do' approach of some students.

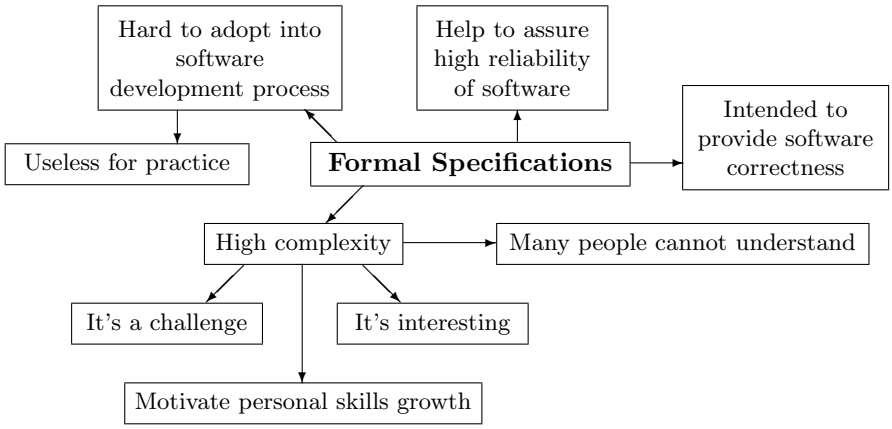

Fig. 3. Example of cluster of 'Formal Specifications' notion.

Some methods that help to change student's approach to new knowledge is based on personal perception, which turns new knowledge into one's own achievement, not only a set of statements taken from books. An example of such a technique is *clustering* [7]. Clustering helps to construct a rich set of associations between the new knowledge and already known things. These associations can be based not only on logical relations, but on emotional attitudes, guesses, or even erroneous assumptions.

Before a lecture on the given topic, students draw clusters of the main notion or statement of the lecture. The following steps are performed.

1. Write the key notion or statement of the lecture in the middle of the blank paper sheet.
2. Write around it, without any plan or ordering, terms or statements that seem to be related to the main notion or expose your personal opinion or attitude to it.

3. Draw links between most closely related terms and statements on the sheet. Links between the main notion and them may not be direct. The following rules should be obeyed.
   - Write everything that occurs, free the imagination and intuition, and try not to use some logical plan.
   - Try to write as many things and draw as many links as you can.
   - Continue to write and draw until you have no more ideas or the time is over.

Fig. 3 shows an example of cluster of *'Formal Specifications'* notion.

Before starting this activity the teacher should clearly say how much time it should take. When students are ready with their clusters, one of them may be asked (but only if he or she wishes) to draw his/her cluster on the blackboard. If there is no one who wishes to do this, the teacher draws his/her own cluster. Others may compare their clusters with the drawn on the blackboard. After the end of the lecture, students may return to their clusters and add or remove something.

Very useful is discussion of clusters in small groups or pairs of students. That may raise questions to be answered on the lecture or on further lessons and makes students more communicative. Further learning is facilitated by associations moulded by the cluster and can be organized as comparison of the information provided by lecturer with those associations.

2.3 Brainstorming

Idea: **Morphological analysis methods for software defect detection**		
Positives	Negatives	Interesting points
Actually new	Source code needed	Can be used in combination with usual static analysis techniques.
Complex defects can be found	Concept of defect should be cleared Complex workaround needed when correct code is considered as a defect	

Fig. 4. Example of subjective evaluation of an idea generated by brainstorm.

For senior students starting research activity to have at the start a rich set of ideas for future elaboration and development is very important. To create such a set of ideas the *brainstorming* [8] technique can be used. The brainstorming session is organized in four phases.

1. On the first phases the people participating in the brainstorming get familiar with the area of student's interest. This part can be organized as follows.

   (a) The student and his/her supervisor prepare a sketch of the future research area, main problems, and methods used in it.
   (b) Each participant reads this sketch.
   (c) Participants formulate a number of questions concerning the problem area. These questions are written on the blackboard to make them accessible for all the participants and the student.
   (d) The student conduct a talk on his/her interests and on an area under consideration. The talk should be organized as general-purpose presentation, but the student tries to answer the questions stated.
   (e) After the talk the questions, which are considered to be answered by the people who posed them, are marked with plus sign. No more questions can be asked.
2. On the second phase the participants suggest possible ideas concerning the future research. This phase is governed by the following rules.
   (a) Each participant can take the floor on half a minute. During this talk only one idea should be formulated.
   (b) The number of talks by one participant is unbounded, but he/she should stop the talk after half a minute if someone else wish to talk.
   (c) The talks should not be interrupted.
   (d) Each participant may not to talk.
   (e) The idea of one talk should be formulated as clear as possible, preferably in one phrase.
   (f) Other ideas should not be questioned, criticized, evaluated or discussed in a talk.
   (g) Expressions "it's obvious", "it's unclear", "you don't understand", "let me explain", "let me refine" are prohibited.
   (h) The rules are enforced by a single host person.
   (i) All the ideas expressed are recorded by the host person.
3. On the third phase the participants are evaluating all the ideas on some subjective, but systematic basis, for example PMI (Plus-Minus-Interesting). The phase can be organized as follows.
   (a) For each idea formulated and each participant a sheet of paper is prepared that contains the idea and three columns for notes. The first column should contain arguments in favor of the idea, the second one – against it, and the third one can contain interesting points of the idea. Another possible scheme – provide pluses, minuses, and implications of the idea.
   (b) Each participant gets the papers on all the ideas and fills them. It is not obligatory to evaluate all the ideas.
   (c) All the sheets are gathered for further integration of assessments. Example of such a sheet is show on Fig. 4.
4. On the last phase the student with the supervisor summarise the evaluation results for all the ideas and reveal the most perspective ones.

Brainstorming session organization should motivate participant to produce interesting ideas, be they fantastic or crazy. It should also make the participants free from authoritative opinions and views, because they have no need to defend their ideas – everything is accepted.

2.4 Debates

One of very useful skills in software development is ability to hear viewpoints different from one's own, to understand the reasons under such a viewpoint, and to take them into consideration. The *debates* [9] technique helps to acquire such a skill. The whole exercise is performed in two steps.

First, the statement to discuss is formulated and recorded on the blackboard. The statement should not be obviously false or true. Then individual participants or groups of participants give arguments in favor or against the statement. Each argument can be supported by additional statements. All the arguments and supporting statements are also recorded on the black board. The goal of all the activity is to give each of the participants the material to form his/her own viewpoint on the topic of the discussion. Another goal is to train their skills in defending their viewpoints and understanding the opposite ones – students know that somebody can object them; they are psychologically ready for that, they are trained to concentrate on the essential matters, to analyse and evaluate different views on the subject, and to respect opposite, maybe unpopular viewpoints.

An example of blackboard records after the discussion of some statement is presented on Fig. 5.

POSIX is a strict standard of operating system functionality

Positives	Negatives
Standard is strict description of something by definition. It presents formal framework where descriptions of many functions are arranged according to formal rules.	Formal rules of document construction do not guarantee the strictness of the contents of text prepared according to these rules.
Support: One can easily note that POSIX requirements are described in rather strict framework.	
POSIX represents a consensus on OS functionality. It exists about 20 years; this is enough time to remove possible defects and to become mature and strict.	POSIX integrates views of many OS vendors that may be rather different.
	Support: Ambiguities are unavoidable in standard supporting several different approaches to OS design.
Many OS architects and developers successfully use POSIX as OS design guide.	
	Take for example description of thread creation function – it does not specify *how* thread creation function interacts with memory management. It says only that this two aspects *may* interact.

Fig. 5. Example of debates.

3 Conclusion

The article presents an idea of integrated learning in technical, cognitive, and social skills needed to use successfully advanced software development techniques, in particular formal methods, in practice. Such integrated learning can be organized with the help of active learning methods. We have presented several techniques that can help to train necessary skills – ability to hear and take into account the opposite viewpoints, ability to communicate effectively on professional topics, and ability to adapt one's knowledge to the current needs.

The four techniques presented are role-playing game, clustering, brainstorming, and debates. Authors successfully used those techniques in training modern model-based testing techniques [10, 11] and are sure that they can be applicable for training in formal methods of any kind, if the result of such training should be actual ability to use them in practice.

Several trainings conducted with use of the techniques presented and some others demonstrate more efficient knowledge acquiring by students and make them more active on their work in the projects.

References

1. A Memorandum on Lifelong Learning: Commission Staff Working Paper. European Commission. Directorate General for Education and Culture. European Commission, 2000. SEC (2000) 1832.
2. S. C. Brock. Practitioners' views on teaching the large introductory college course. 12 pp. Kansas State University Center for Faculty Evaluation and Development. ERIC Document Reproduction Service No. ED 171 208, 1976.
3. Labinowicz, Ed. The Piaget Primer: Thinking, Learning, Teaching. Menlo Park, CA: Addison-Wesley Publishing Co., 1980.
4. Pedagogical technologies. Ed. V. S. Kukushkin, Moscow, 2004 (in Russian).
5. Dennis M. Adams. Simulation Games: An Approach to Learning. Worthington, Ohio, Charles A. Jones Publishing Company, 1973.
6. John P. Hertel, and Barbara J. Mills. Using Simulations to Promote Learning in Higher Education: An Introduction. Stylus Publishing, Herndon, VA, 2002.
7. Tony Buzan. Use Both Sides of Your Brain. New York: Dutton, 1974.
8. James L. Marra. Advertising Creativity: Techniques For Generating Ideas. Englewood Cliffs, NJ: Prentice Hall, 1990.
9. Ronald T. Hyman. Improving Discussion Leadership. New York: Columbia Univ., Teachers College Press, 1980.
10. O. L. Petrenko, V. A. Omelchenko. Rapid Trainings on Specification Based Testing Tools. In Proc. of 1-st South-East European Workshop on Formal Methods, Thessaloniki, Greece, 2003.
11. V. V. Kuliamin, V. A. Omeltchenko, O. L. Petrenko. Learning advanced software development methods: problems and solutions. Proceedings of ISP RAS, v. 5, 2004, pp. 91–108.